Lecture 5
Least-squares

- least-squares (approximate) solution of overdetermined equations
- projection and orthogonality principle
- least-squares estimation
- BLUE property

Overdetermined linear equations

consider $y = Ax$ where $A \in \mathbf{R}^{m \times n}$ is (strictly) skinny, *i.e.*, $m > n$

- called *overdetermined* set of linear equations (more equations than unknowns)
- for most y, cannot solve for x

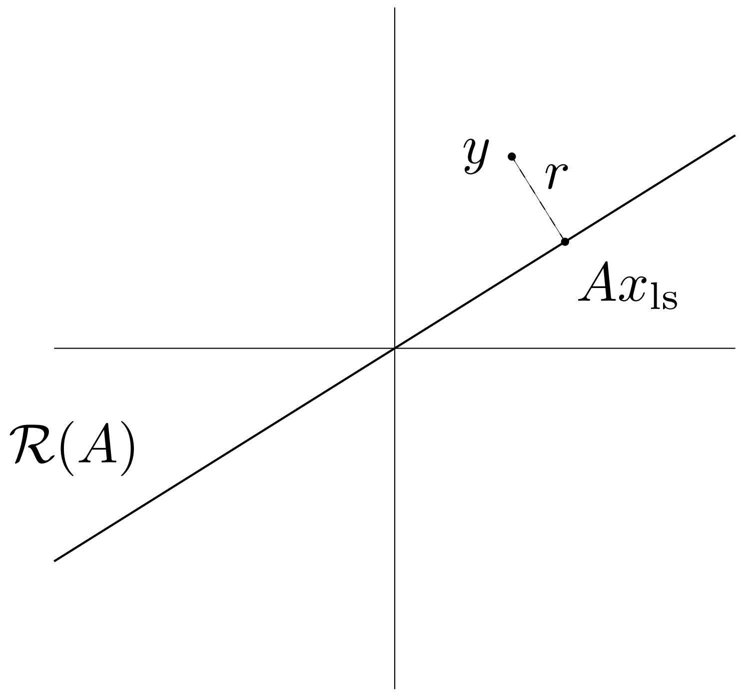
one approach to *approximately* solve $y = Ax$:

- define *residual* or error $r = Ax - y$
- find $x = x_{\mathrm{ls}}$ that minimizes $\|r\|$

x_{ls} called *least-squares* (approximate) solution of $y = Ax$

Geometric interpretation

$Ax_{\rm ls}$ is point in $\mathcal{R}(A)$ closest to y ($Ax_{\rm ls}$ is *projection* of y onto $\mathcal{R}(A)$)



Least-squares (approximate) solution

- assume A is full rank, skinny
- to find x_{ls}, we'll minimize norm of residual squared,

$$\|r\|^2 = x^T A^T A x - 2y^T A x + y^T y$$

- set gradient w.r.t. x to zero:

$$\nabla_x \|r\|^2 = 2A^T A x - 2A^T y = 0$$

- yields the *normal equations*: $A^T A x = A^T y$
- assumptions imply $A^T A$ invertible, so we have

$$x_{\text{ls}} = (A^T A)^{-1} A^T y$$

. . . a very famous formula

- $x_{\rm ls}$ is linear function of y

- $x_{\rm ls} = A^{-1}y$ if A is square

- $x_{\rm ls}$ solves $y = Ax_{\rm ls}$ if $y \in \mathcal{R}(A)$

- $A^\dagger = (A^TA)^{-1}A^T$ is called the *pseudo-inverse* of A

- $A^\dagger$ is a *left inverse* of (full rank, skinny) A:

$$A^\dagger A = (A^TA)^{-1}A^T\, A = I$$

Projection on $\mathcal{R}(A)$

$Ax_{\rm ls}$ is (by definition) the point in $\mathcal{R}(A)$ that is closest to y, $i.e.$, it is the *projection* of y onto $\mathcal{R}(A)$

$$Ax_{\rm ls} = \mathcal{P}_{\mathcal{R}(A)}(y)$$

- the projection function $\mathcal{P}_{\mathcal{R}(A)}$ is linear, and given by

$$\mathcal{P}_{\mathcal{R}(A)}(y) = Ax_{\rm ls} = A(A^TA)^{-1}A^Ty$$

- $A(A^TA)^{-1}A^T$ is called the *projection matrix* (associated with $\mathcal{R}(A)$)

Orthogonality principle

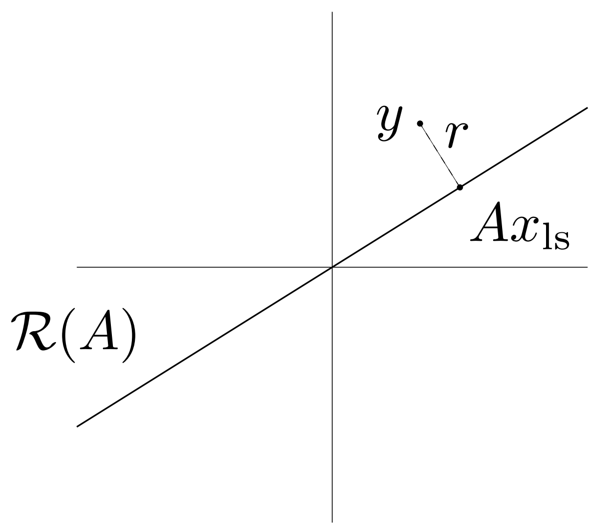
optimal residual

$$r = Ax_{\rm ls} - y = (A(A^TA)^{-1}A^T - I)y$$

is orthogonal to $\mathcal{R}(A)$:

$$\langle r, Az\rangle = y^T(A(A^TA)^{-1}A^T - I)^T Az = 0$$

for all $z \in \mathbf{R}^n$



Completion of squares

since $r = Ax_{\rm ls} - y \perp A(x - x_{\rm ls})$ for any x, we have

$$\begin{aligned} \|Ax - y\|^2 &= \|(Ax_{\rm ls} - y) + A(x - x_{\rm ls})\|^2 \\ &= \|Ax_{\rm ls} - y\|^2 + \|A(x - x_{\rm ls})\|^2 \end{aligned}$$

this shows that for $x \neq x_{\rm ls}$, $\|Ax - y\| > \|Ax_{\rm ls} - y\|$

Least-squares via QR factorization

- $A \in \mathbf{R}^{m \times n}$ skinny, full rank
- factor as $A = QR$ with $Q^T Q = I_n$, $R \in \mathbf{R}^{n \times n}$ upper triangular, invertible
- pseudo-inverse is

$$(A^T A)^{-1} A^T = (R^T Q^T Q R)^{-1} R^T Q^T = R^{-1} Q^T$$

so $x_{\mathrm{ls}} = R^{-1} Q^T y$

- projection on $\mathcal{R}(A)$ given by matrix

$$A(A^T A)^{-1} A^T = A R^{-1} Q^T = Q Q^T$$

Least-squares via full QR factorization

- full QR factorization:

$$A = [Q_1\ Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

with $[Q_1\ Q_2] \in \mathbf{R}^{m \times m}$ orthogonal, $R_1 \in \mathbf{R}^{n \times n}$ upper triangular, invertible

- multiplication by orthogonal matrix doesn't change norm, so

$$\begin{aligned} \|Ax - y\|^2 &= \left\| [Q_1\ Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - y \right\|^2 \\ &= \left\| [Q_1\ Q_2]^T [Q_1\ Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - [Q_1\ Q_2]^T y \right\|^2 \end{aligned}$$

$$
\begin{aligned}
&= \left\| \begin{bmatrix} R_1 x - Q_1^T y \\ -Q_2^T y \end{bmatrix} \right\|^2 \\
&= \|R_1 x - Q_1^T y\|^2 + \|Q_2^T y\|^2
\end{aligned}
$$

- this is evidently minimized by choice $x_{\mathrm{ls}} = R_1^{-1} Q_1^T y$ (which makes first term zero)

- residual with optimal x is

$$
A x_{\mathrm{ls}} - y = -Q_2 Q_2^T y
$$

- $Q_1 Q_1^T$ gives projection onto $\mathcal{R}(A)$

- $Q_2 Q_2^T$ gives projection onto $\mathcal{R}(A)^\perp$

Least-squares estimation

many applications in inversion, estimation, and reconstruction problems have form

$$y = Ax + v$$

- x is what we want to estimate or reconstruct
- y is our sensor measurement(s)
- v is an unknown *noise* or *measurement error* (assumed small)
- ith row of A characterizes ith sensor

least-squares estimation: choose as estimate $\hat{x}$ that minimizes

$$\|A\hat{x} - y\|$$

$i.e.$, deviation between

- what we actually observed (y), and
- what we would observe if $x = \hat{x}$, and there were no noise ($v = 0$)

least-squares estimate is just $\hat{x} = (A^T A)^{-1} A^T y$

BLUE property

linear measurement with noise:

$$y = Ax + v$$

with A full rank, skinny

consider a *linear estimator* of form $\hat{x} = By$

- called *unbiased* if $\hat{x} = x$ whenever $v = 0$
  ($i.e.$, no estimation error when there is no noise)

  same as $BA = I$, $i.e.$, B is left inverse of A

- estimation error of unbiased linear estimator is

$$x - \hat{x} = x - B(Ax + v) = -Bv$$

obviously, then, we'd like B 'small' (and $BA = I$)

- **fact:** $A^\dagger = (A^TA)^{-1}A^T$ is the *smallest* left inverse of A, in the following sense:

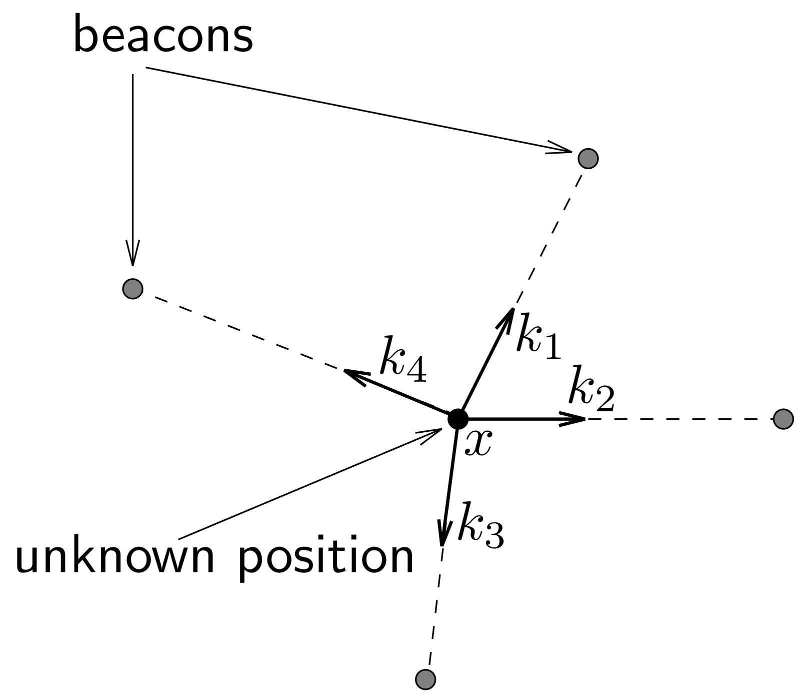
for any B with $BA = I$, we have

$$\sum_{i,j} B_{ij}^2 \geq \sum_{i,j} A_{ij}^{\dagger 2}$$

$i.e.$, least-squares provides the *best linear unbiased estimator* (BLUE)

Navigation from range measurements

navigation using range measurements from *distant* beacons



beacons far from unknown position $x \in \mathbf{R}^2$, so linearization around $x = 0$ (say) nearly exact ranges $y \in \mathbf{R}^4$ measured, with measurement noise v:

$$y = -\begin{bmatrix} k_1^T \\ k_2^T \\ k_3^T \\ k_4^T \end{bmatrix} x + v$$

where k_i is unit vector from 0 to beacon i

measurement errors are independent, Gaussian, with standard deviation 2 (details not important)

problem: estimate $x \in \mathbf{R}^2$, given $y \in \mathbf{R}^4$

(roughly speaking, a $2\!:\!1$ measurement redundancy ratio)

actual position is $x = (5.59, 10.58)$;
measurement is $y = (-11.95, -2.84, -9.81, 2.81)$

Just enough measurements method

y_1 and y_2 suffice to find x (when $v = 0$)

compute estimate $\hat{x}$ by inverting top (2×2) half of A:

$$\hat{x} = B_{\mathrm{je}} y = \begin{bmatrix} 0 & -1.0 & 0 & 0 \\ -1.12 & 0.5 & 0 & 0 \end{bmatrix} y = \begin{bmatrix} 2.84 \\ 11.9 \end{bmatrix}$$

(norm of error: 3.07)

Least-squares method

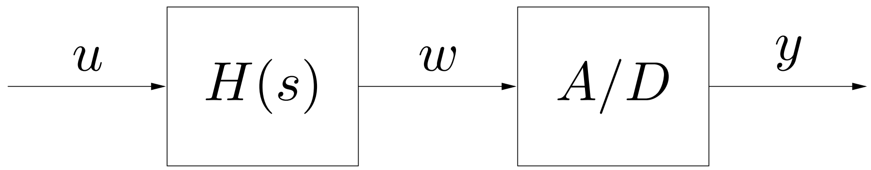
compute estimate $\hat{x}$ by least-squares:

$$\hat{x} = A^\dagger y = \begin{bmatrix} -0.23 & -0.48 & 0.04 & 0.44 \\ -0.47 & -0.02 & -0.51 & -0.18 \end{bmatrix} y = \begin{bmatrix} 4.95 \\ 10.26 \end{bmatrix}$$

(norm of error: 0.72)

- $B_{\rm je}$ and $A^\dagger$ are both left inverses of A
- larger entries in B lead to larger estimation error

Example from overview lecture



- signal u is piecewise constant, period $1\,\mathrm{sec}$, $0 \leq t \leq 10$:

$$u(t) = x_j, \quad j-1 \leq t < j, \quad j = 1, \ldots, 10$$

- filtered by system with impulse response $h(t)$:

$$w(t) = \int_0^t h(t-\tau)u(\tau)\, d\tau$$

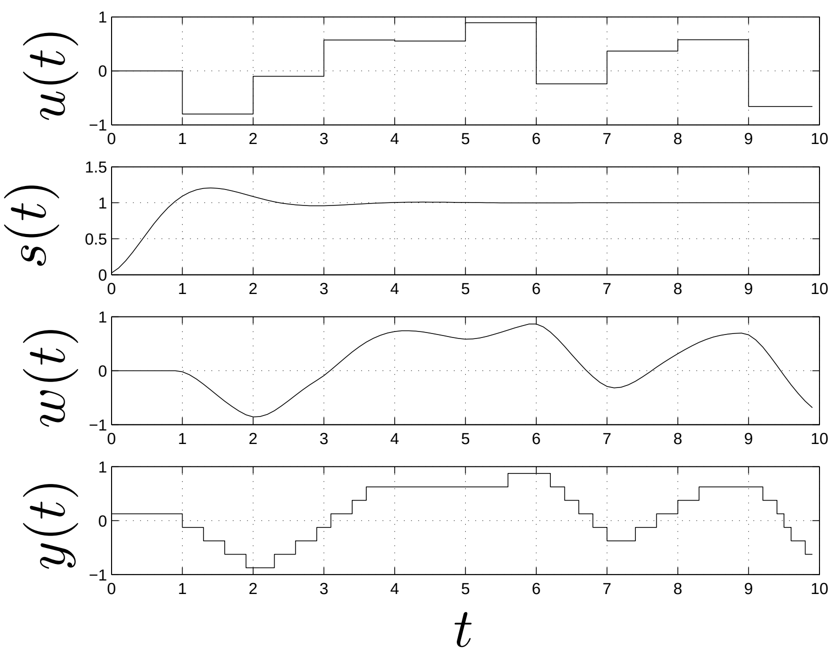
- sample at 10Hz: $\tilde{y}_i = w(0.1i)$, $i = 1, \ldots, 100$

- 3-bit quantization: $y_i = Q(\tilde{y}_i)$, $i = 1, \ldots, 100$, where Q is 3-bit quantizer characteristic

$$Q(a) = (1/4)\left(\mathbf{round}(4a + 1/2) - 1/2\right)$$

- **problem:** estimate $x \in \mathbf{R}^{10}$ given $y \in \mathbf{R}^{100}$

example:

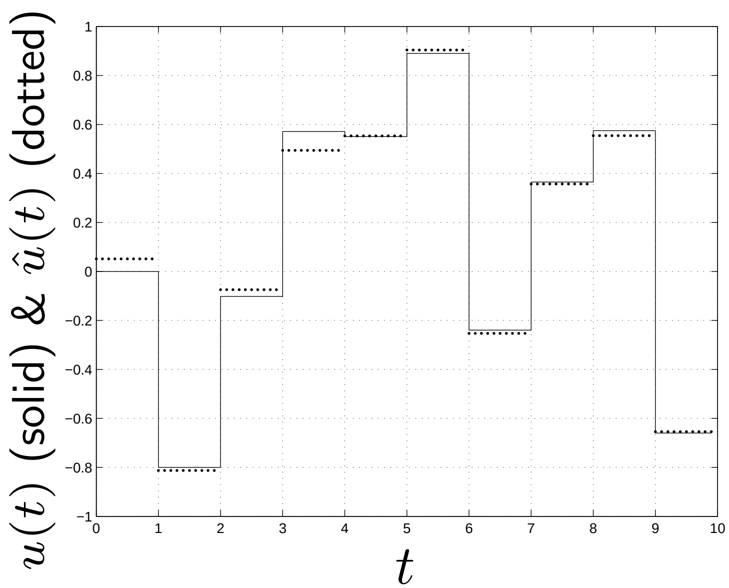


we have $y = Ax + v$, where

- $A \in \mathbf{R}^{100\times 10}$ is given by $A_{ij} = \displaystyle\int_{j-1}^{j} h(0.1i - \tau)\, d\tau$

- $v \in \mathbf{R}^{100}$ is *quantization error:* $v_i = Q(\tilde{y}_i) - \tilde{y}_i$ (so $|v_i| \leq 0.125$)

least-squares estimate: $x_{\rm ls} = (A^TA)^{-1}A^Ty$

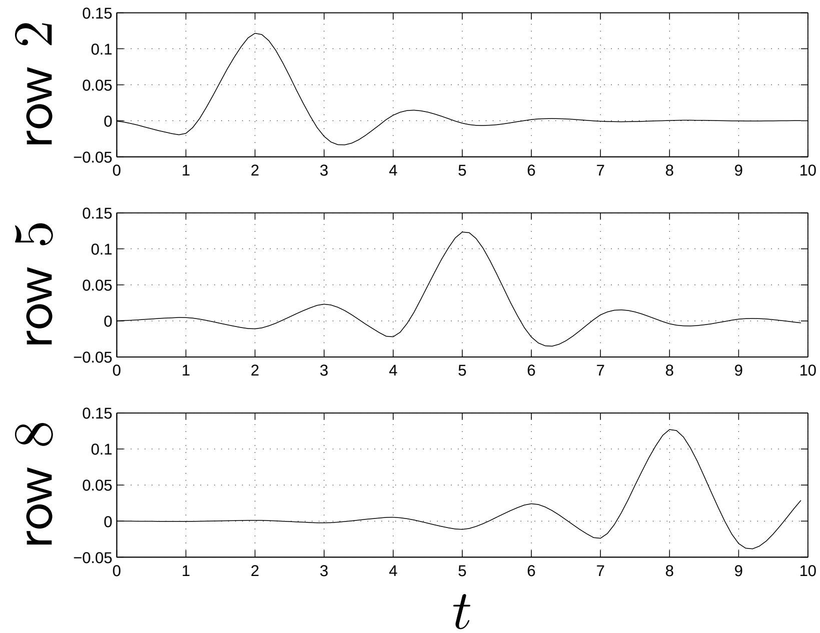


RMS error is $\dfrac{\|x - x_{\rm ls}\|}{\sqrt{10}} = 0.03$

better than if we had no filtering! (RMS error 0.07)

more on this later . . .

some rows of $B_{\rm ls} = (A^T A)^{-1} A^T$:



- rows show how sampled measurements of y are used to form estimate of x_i for $i = 2, 5, 8$
- to estimate x_5, which is the original input signal for $4 \le t < 5$, we mostly use $y(t)$ for $3 \le t \le 7$